A Summary of notation and important matrices

The prefix-sum linear operator $\mathbf{S}$, and its inverse:

$$\mathbf{S} := \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{S}^{-1} := \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}. \tag{10}$$

Representation of momentum SGD as a linear operator $\mathbf{M} = \mathbf{M}^{(\eta)}\mathbf{M}^{(\beta)}$:

$$\mathbf{M}^{(\eta)} := \begin{pmatrix} \eta_1 & 0 & 0 & \cdots & 0 \\ \eta_1 & \eta_2 & 0 & \cdots & 0 \\ \eta_1 & \eta_2 & \eta_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \eta_1 & \eta_2 & \eta_3 & \cdots & \eta_n \end{pmatrix} \quad \text{and} \quad \mathbf{M}^{(\beta)} := \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \beta & 1 & 0 & \cdots & 0 \\ \beta^2 & \beta & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta^{n-1} & \beta^{n-2} & \beta^{n-3}\cdots & 1 & \end{pmatrix} \tag{11}$$

Summary of notation The following table briefly summarizes notation used throughout this work.

$\mathbf{g}_i \in \mathbb{R}^d$	Input (e.g. gradient) on step i of the online process.
$\mathbf{G} \in \mathbb{R}^{n \times d}$	Matrix of all inputs, $\mathbf{g}_i = \mathbf{G}_{[i,:]}$.
$\mathbf{A} \in \mathbb{R}^{n \times n}$	Lower-triangular linear query matrix to be factorized as $\mathbf{A} = \mathbf{BC}$.
$\lambda_{\min}(\mathbf{A})$, $\lambda_{\max}(\mathbf{A})$.	Smallest and largest eigenvalues of real matrix $\mathbf{A}$.
$\mathbf{A}^*$	Conjugate transpose of $\mathbf{A}$.
$\mathbf{X}^\star$	A matrix $\mathbf{X}$ that is "optimal" in a context-dependent sense.
$\mathbf{A}^\dagger$	Moore-Penrose pseudoinverse of matrix $\mathbf{A}$.
$\mathbf{A}_{[i,j]}$	The $(i,j)^{\text{th}}$ entry of matrix $\mathbf{A}$.
$\mathbf{A}_{[i,:]}$ and $\mathbf{A}_{[:,j]}$	The i^{th} row and j^{th} column.

B Future work.

Each of the sections above poses a unique set of problems for future investigation, many interrelated. We will highlight only some of the major questions left open by this work.

Scalable mechanism implementations Theorem 2.1 shows that we need not restrict ourselves to any particular matrix structure in order to guarantee privacy over adaptive streams. Appendix H shows we can find efficient approximations for the case of prefix sums, but this leaves open the question of whether better or more general approximations are possible, or whether one can optimize over structures that allow efficient implementations directly.

Analysis and numerics of ϕ Theorem 3.3 represents a usable convergence result for iterates of the mapping ϕ; on the other hand, it represents only partial progress on the conjecture of global convergence of these iterates. Though we factorized many distinct matrices in the course of writing this paper, we generated no reason to doubt this conjecture. Indeed, the speed of convergence of these iterates of ϕ (see Appendix E.4) only makes this method more intriguing from a theoretical perspective. Further, though the fixed-point method utilized to compute these factorizations has enabled significant exploration (as detailed in Section 4), it still does not quite represent the optimal algorithm for computing these optima: an explicit formula for the fixed point of ϕ would clearly be desirable, and might yield interesting insights into the structure of these optimal matrices.

We finally note that for production use, additional care will be needed to ensure that claimed privacy guarantees fully account for floating point imprecision.

Adaptive choice of the query While the sequence of gradients during optimization is adaptive (subsequent gradients depend on previous gradients), as we have seen SGD with momentum can be expressed as a fixed linear operator $\mathbf{M}$. Data-independent learning rate schedules can be incorporated into an optimization matrix in a similar fashion, again allowing for optimal DP matrix mechanisms. However, adaptive learning rate schedules such as AdaGrad amount to a *non-linear* (and adaptive, not fixed) map on the gradient sequence; hence a very interesting open question is to see if the approach used here can be extended to adaptive optimization algorithms.

C Tree aggregation and decoding as matrix factorization

As mention in Section 1, the tree data structure $\mathcal{T}$ is linear in the data matrix $\mathbf{G}$ (all of its internal nodes are linear combinations of the rows $\mathbf{G}$). Therefore the mapping $\mathbf{G} \to \mathcal{T}$ can be represented as multiplication by a matrix. We present a simple recursive construction of this matrix. The base case is the 1×1 matrix $[1]$, which we will denote by $\mathbf{C}_{\mathcal{T}}^{(1)}$; we will define $\mathbf{C}_{\mathcal{T}}^{(k)} \in \mathbb{R}^{(2^k-1)\times(2^{k-1})}$ to be the matrix constructed by duplicating $\mathbf{C}_{\mathcal{T}}^{(k-1)}$ on the diagonal, and adding one more row of constant 1s. That is,

$$\mathbf{C}_{\mathcal{T}}^{(1)} := (1)\,, \mathbf{C}_{\mathcal{T}}^{(2)} := \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}, \mathbf{C}_{\mathcal{T}}^{(3)} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \tag{12}$$

and so on. Each row of $\mathbf{C}_{\mathcal{T}}^{(k)}\mathbf{G}$ can be seen readily to correspond to a node of the binary tree $\mathcal{T}$ constructed from $\mathbf{G}$, assuming $n = 2^{k-1}$ (possibly padding with zeros if needed).

With this construction, it is straightforward to represent both vanilla differentially-private binary tree aggregation and the Honaker variant as instantiations of the matrix factorization framework. For a vector $\mathbf{x}$ with $n = 2^{k-1}$ entries, vanilla binary-tree aggregation can be represented as $\mathbf{C} = \mathbf{C}_{\mathcal{T}}^{(k)}$, $\mathbf{B}$ an appropriate $\{0, 1\}$-valued matrix satisfying $\mathbf{BC} = \mathbf{S}$ for prefix-sum $\mathbf{S}$. The Honaker estimators can both be computed as (real-valued) matrices also satisfying $\mathbf{BC} = \mathbf{S}$, and are in fact optimal:

Proposition C.1. *For the prefix-sum matrix* $\mathbf{S}$ *with* $n = 2^{k-1}$ *rows, the (non-streaming) Honaker fully efficient estimator represents the minimal-loss factorization for prefix sum* $\mathbf{S} = \mathbf{BC}$ *for* $\mathbf{C} = \mathbf{C}_{\mathcal{T}}^{(k)}$*. This estimator is precisely* $\mathbf{SC}^{\dagger}$*. The streaming Honaker estimator-from-below represents the minimal loss factorization satisfying the property that the* j^{th} *row of* $\mathbf{B}$ *zeros out rows in the matrix* $\mathbf{CG}$ *which place nonzero weight on the* i^{th} *row of* $\mathbf{G}$ *for* $i > j$*. The Honaker estimator-from-below can be expressed similarly row-by-row with a constrained pseudoinverse of* $\mathbf{C}$.

Proof. We begin by recalling a geometric property of the Moore-Penrose pseudoinverse. Theorem 2.1.1 of [38] states that for any matrix $\mathbf{C} \in \mathbb{C}^{m\times n}$, vector $\mathbf{s} \in \mathbb{C}^m$, the vector $\mathbf{C}^{\dagger}\mathbf{s}$ is the minimal least-squares solution to the linear system $\mathbf{Cx} = \mathbf{s}$. Notice that this statement is implicitly a statement of uniqueness; $\mathbf{C}^{\dagger}\mathbf{s}$ is the *unique* minimal-norm solution to $\mathbf{Cx} = \mathbf{s}$, assuming feasability of this equation. Since the square of the Frobenius norm of the matrix $\mathbf{B}$ is the sum of the squared norms of its rows, we may apply this Theorem row-by-row to $\mathbf{B}$ to demonstrate that the minimal Frobenius norm solution $\mathbf{B}$ to $\mathbf{S} = \mathbf{BC}$ for fixed $\mathbf{C}$ is $\mathbf{SC}^{\dagger}$.

This minimal Frobenius norm property may be translated to a statistical perspective. That is, for a fixed matrix $\mathbf{C}$ and data matrix $\mathbf{G}$, $\mathbf{SC}^{\dagger}$ represents the minimal-variance unbiased linear estimator for $\mathbf{SG}$ given the noisy estimates $\mathbf{CG} + \mathbf{Z}$. This is precisely the definition of Honaker's fully efficient estimator in Section 3.4 of [22], and we have the first statement of this proposition.

The second follows similarly, but leveraging instead the geometric properties of the constrained pseudoinverse. These properties are collected in Theorem 3.6.3 of [38], and allow us to compute directly the optimal $\mathbf{B}$ under constraints that certain entries in each row must be 0, corresponding to the constraints stated in the proposition. By construction of the matrices $\mathbf{C}_{\mathcal{T}}^{(k)}$, the property described in the statement of Proposition C.1 corresponds to restricting the linear estimator computed from a binary tree to depend only on the information below the nodes corresponding to the 1s in a binary expansion of the index of the partial sum under consideration. This is precisely the definition of the estimator from below in Section 3.2 of [22]. □

D Proofs and missing details for Section 2

Proof of Proposition 2.1. The key idea is that the nonadaptive version of the definition implies a bound on the log-odds ratio that always holds (even after the fact).

For simplicity, we focus on the case where the universe of possible outputs $\mathbf{a}$ is discrete (to avoid measurability issues).

Fix an adversary $\mathcal{A}$ and mechanism $\mathcal{M}$. Recall side is fixed an unknown to the adversary. When $\mathsf{side} = 0$, the probability of a particular view $(\mathbf{G}, \mathbf{H}, \mathbf{a})$ is the following. We write $(\mathbf{G}, \mathbf{H}, \mathbf{a}) \leftarrow \langle \mathcal{M}, \mathcal{A} \rangle_0$ for the event with sequence of mechanism outputs $\mathbf{a}$, when the mechanism and the adversary are operating with the variable $\mathsf{side} = 0$, and the neighboring data streams are $\mathbf{G}$ and $\mathbf{H}$ (and analogously for $\mathsf{side} = 1$).

$$
\begin{array}{rcl}
\Pr((\mathbf{G},\mathbf{H},\mathbf{a}) \leftarrow \langle\mathcal{M},\mathcal{A}\rangle_0) = & & \\
\Pr\left(\mathcal{A}() = (\mathbf{g}_1,\mathbf{h}_1)\right) & \times & \Pr\left(\mathcal{M}(\mathbf{g}_1)=\mathbf{a}_1\right) \times \\
\Pr\left(\mathcal{A}(\mathbf{a}_1) = (\mathbf{g}_2,\mathbf{h}_2) \middle| \mathbf{g}_1,\mathbf{h}_1\right) & \times & \Pr\left(\mathcal{M}(\mathbf{g}_2)=\mathbf{a}_2 \middle| \mathbf{g}_1,\mathbf{a}_1\right) \times \\
 & \cdots & \\
\underbrace{\Pr\left(\mathcal{A}(\mathbf{a}_{n-1}) = (\mathbf{g}_n,\mathbf{h}_n) \middle| \mathbf{g}_1,\ldots,\mathbf{g}_{n-1},\mathbf{h}_1,\ldots,\mathbf{h}_{n-1}\right)}_{\text{these do not depend on } \mathsf{side}} & \times & \underbrace{\Pr\left(\mathcal{M}(\mathbf{g}_n)=\mathbf{a}_n \middle| \mathbf{g}_1,\ldots,\mathbf{g}_{n-1},\mathbf{a}_1,\ldots,\mathbf{a}_{n-1}\right)}_{\text{these terms depend on } \mathsf{side}} .
\end{array}
$$

The probability of $(\mathbf{G}, \mathbf{H}, \mathbf{a})$ when $\mathsf{side} = 1$ is similar, except that the inputs to $\mathcal{M}$ are now $\mathbf{h}_t$'s instead of $\mathbf{g}_t$'s. Either way, we get a product of $2n$ terms, half of which are about the probability of $\mathcal{A}$'s outputs, and half of which are about $\mathcal{M}$'s outputs. They key fact here is that the terms describing $\mathcal{A}$'s output are the same in both expressions. When we take the ratio, therefore, those terms cancel out and we obtain:

$$
\begin{aligned}
&\frac{\Pr((\mathbf{G},\mathbf{H},\mathbf{a}) \leftarrow \langle\mathcal{M},\mathcal{A}\rangle_0)}{\Pr((\mathbf{G},\mathbf{H},\mathbf{a}) \leftarrow \langle\mathcal{M},\mathcal{A}\rangle_1)} \\
&= \frac{\Pr\left(\mathcal{M}(\mathbf{g}_1)=\mathbf{a}_1\right) \times \Pr\left(\mathcal{M}(\mathbf{g}_2)=\mathbf{a}_2 \middle| \mathbf{g}_1,\mathbf{a}_1\right) \times \cdots \times \Pr\left(\mathcal{M}(\mathbf{g}_n)=\mathbf{a}_n \middle| \mathbf{g}_1,\ldots,\mathbf{g}_{n-1},\mathbf{a}_1,\ldots,\mathbf{a}_{n-1}\right)}{\Pr\left(\mathcal{M}(\mathbf{h}_1)=\mathbf{a}_1\right) \times \Pr\left(\mathcal{M}(\mathbf{h}_2)=\mathbf{a}_2 \middle| \mathbf{h}_1,\mathbf{a}_1\right) \times \cdots \times \Pr\left(\mathcal{M}(\mathbf{h}_n)=\mathbf{a}_n \middle| \mathbf{h}_1,\ldots,\mathbf{h}_{n-1},\mathbf{a}_1,\ldots,\mathbf{a}_{n-1}\right)} \\
&= \frac{\Pr(\mathcal{M}(\mathbf{g}_1,\ldots,\mathbf{g}_n) = (\mathbf{a}_1,\ldots,\mathbf{a}_n))}{\Pr(\mathcal{M}(\mathbf{h}_1,\ldots,\mathbf{h}_n) = (\mathbf{a}_1,\ldots,\mathbf{a}_n))} .
\end{aligned}
$$

This last expression involves no adversary—it is simply the ratio of the probabilities that the mechanism would have produced a given output sequence if the sequences $\mathbf{G}$ and $\mathbf{H}$ had been specified *nonadaptively*. Since $\mathbf{G}, \mathbf{H}$ are always valid neighboring sequences, and since the nonadaptive mechanism's guarantee holds for all output sequences, the ratio above is bounded between $e^{-\varepsilon}$ and e^{ε}, as desired.

□

Proof of Theorem 2.1. The idea is to show that, when $\mathbf{A}$ is lower triangular, the mechanism $\mathcal{M}$ can be rewritten in such a way that the adaptive privacy of $\mathcal{M}$ can be deduced from the privacy guarantees of the usual Gaussian mechanism with adaptively selected queries.

Let $(\mathbf{L}, \mathbf{Q})$ form a lower-triangular LQ-factorization of $\mathbf{B}$, meaning that $\mathbf{L}$ is lower triangular, $\mathbf{Q}$ is orthonormal, and $\mathbf{B} = \mathbf{LQ}$. By assumption, $\mathbf{A}$ is square and invertible, so $\mathbf{L}$ and $\mathbf{Q}$ are also square and invertible. Now consider the modified mechanism

$$\tilde{\mathcal{M}}(\mathbf{G}) = \mathbf{L}(\mathbf{QCG} + \mathbf{Z}) \quad \text{where } \mathbf{Z} \sim \mathcal{N}(0, \kappa^2\sigma^2)^{n \times d}$$

where $\kappa\sigma$ is the same noise standard deviation as in the original mechanism. Since $\mathbf{L}$ and $\mathbf{A}$ are lower triangular, it also means that $\mathbf{L}^{-1}\mathbf{A} = \mathbf{QC}$ is also lower triangular. This means that $\mathbf{QCG} + \mathbf{Z}$ can operate in the continuous release model, as row i of $\mathbf{QCG}$ depends only on the first i rows of $\mathbf{G}$.

Next, we further show the mechanism $\mathbf{QCG}+\mathbf{Z}$ (that is, $\tilde{\mathcal{M}}$ without the post-processing operation of multiplying with $\mathbf{L}$) is an instance of the standard Gaussian mechanism for computing an adaptively defined function *in the continuous release model* with a guaranteed bound on the global ℓ_2 sensitivity.[6] Let $\mathbf{G}, \mathbf{H} \in \mathcal{N}$ be any two *fixed* neighboring data streams with $\|\mathbf{C}(\mathbf{G}-\mathbf{H})\|_F \leq \kappa$. Then because $\mathbf{Q}$ is orthonormal we have $\|\mathbf{QC}(\mathbf{G}-\mathbf{H})\|_F \leq \kappa$. Letting $\mathbf{g} = \text{flatten}(\mathbf{QCG}) \in \mathbb{R}^{nd}$ and $\mathbf{h} = \text{flatten}(\mathbf{QCH}) \in \mathbb{R}^{nd}$, we have $\|\mathbf{g}-\mathbf{h}\|_2 \leq \kappa$. Hence, $\mathbf{QCG}+\mathbf{Z}$ is equivalent to an application of the standard Guassian mechanism on inputs $\mathbf{QCG}$, and the result for adaptive streams follows from Claim D.1 below. This claim holds as the privacy loss random variable is stochastically dominated by an appropriate normally-distributed random variable (e.g., [56]).

Claim D.1 (Folklore)**.** *Consider a streaming data vector* $\mathbf{g} = [g_1, \ldots, g_n] \in \mathbb{R}^n$ *s.t. for any neighboring stream* $\mathbf{h}$ *we have the* ℓ_2*-sensitivity* $\|\mathbf{g}-\mathbf{h}\|_2 \leq \kappa$*. If* $\mathbf{g}+\mathcal{N}(0, \kappa^2\sigma^2)^n$ *satisfy* (ε, δ)*-DP (or* ρ*-zCDP or* μ*-Gaussian DP) in the nonadaptive continual release model, then the same mechanism satisfies the same privacy guarantee in the adaptive continuous release model.*

As $\mathbf{L}$ is lower-triangular, the adaptive streaming DP guarantee of $\mathbf{QCG}+\mathbf{Z}$ extends to the mechanism $\tilde{\mathcal{M}}$ by the post-processing property of DP.

Finally, we have

$$\mathcal{M}(\mathbf{G}) = \mathbf{B}(\mathbf{CG}+\mathbf{Z}) = \mathbf{L}(\mathbf{QCG}+\mathbf{QZ}),$$

and so the only difference from $\tilde{\mathcal{M}}$ is the use of noise $\mathbf{QZ}$ vs $\mathbf{Z}$. Since $\mathbf{Q}$ is orthonormal and the Gaussian distribution is rotationally invariant, $\mathbf{QZ}$ and $\mathbf{Z}$ are identically distributed, and hence $\mathcal{M}$ and $\tilde{\mathcal{M}}$ produce identical output distributions on any fixed data set $\mathbf{G}$. Thus an adversary can simulate $\mathcal{M}$ given access to $\tilde{\mathcal{M}}$. This in turn means that the privacy guarantee of the mechanism $\tilde{\mathcal{M}}$ transfers to the mechanism $\mathcal{M}$. This completes the proof. ☐

Proof of Proposition 2.2. The existence of such a lower-triangular factorization with identical induced matrix mechanism distribution follows directly from the proof of Theorem 2.1. The body of the proof leverages the distributional equivalence of all mechanisms expressible as

$$(\mathbf{BU})\left(\mathbf{U}^*\mathbf{C}\right).$$

Since $\mathbf{A}$ is lower-triangular, letting $\mathbf{U} = \mathbf{R}^*$ recovers a lower-triangular mechanism (IE, both terms in the factorization are lower-triangular) which is distributionally equivalent to the factorization $\mathbf{A} = \mathbf{BC}$.

The claimed uniqueness follows from uniqness of the QR factorization with all-nonnegative diagonal entries; see, e.g., [57, Theorem 2.1.14]. ☐

D.1 Not all additive noise mechanisms are adaptively private

Consider the following two step sampling process[7]:

1. $A \sim \mathcal{N}(0, \mathbb{I}_d)$.
2. $B \sim \mathcal{N}(0, \Sigma)$ where $\Sigma = \mathbb{I} - (1-\eta)\frac{AA^t}{\|A\|^2}$ and $\eta = 5\frac{\sqrt{\log(d)}}{d}$. Observe that, conditioned on $A = a$, we can write B as the sum of independent random variables $B = B_1 + B_2$ where $B_1 \sim \mathcal{N}(0, \mathbb{I} - \frac{aa^t}{\|a\|^2})$—a component orthogonal to a with variance 1 in all other directions—and $B_2 \sim \eta\frac{a}{\|a\|} \cdot \mathcal{N}(0,1)$—a component parallel to a with much smaller variance η in that direction.
3. Return (A, B)

---

[6] Observe the same claim cannot be made for $\mathcal{M}$, e.g., as $\mathbf{CG}+\mathbf{Z}$ cannot be used in the continuous release setting as in general $\mathbf{C}$ induces a dependence on not-yet-seen data.

[7] For the clarity of noation, in this section we will refer to random variables with uppercase, and their corresponding instantiations with lower case. Additionally, all norms are $\|\cdot\|_2$.

Now consider a mechanism $\mathcal{M}$ that takes a parameter σ and two inputs of the form $(x_1, x_2) \in (\mathbb{R}^d)^2$ where $x_1 = 0$ (always) and x_2 has Euclidean norm at most 1, and returns $(x_1 + \sigma A, x_2 + \sigma B)$. The mechanism can be run interactively, in which case x_2 could be selected based on A, which can be deduced from $x_1 + \sigma A$. The notion of neighboring here is trivial: every pair $(0, x_2)$ is a neighbor of every other pair $(0, y_2)$ so long as $\|x_2\|$ and $\|y_2\|$ are at most 1.

For simplicity, we formulate the mechanism for the special case when $n = 2$ and the first input is forced to be 0, but similar constructions and reasoning apply for larger n and other types of inputs.

Proposition D.1. *$\mathcal{M}$ is nonadaptively (ε, δ)-DP with parameters $\varepsilon = \Theta(\sqrt{\ln(1/\delta)}/\sigma)$ when d is sufficiently large and $\delta \geq \exp(-cd)$ for an absolute constant $c > 0$.*

Proof. Let $(0, x_2)$ and $(0, y_2)$ be the inputs submitted by the adversary. Let $W = \langle x_2 - y_2, \frac{A}{\|A\|}\rangle$. Observe that $\langle x_2 - y_2, A\rangle$ distributed as $N(0, \|x_2 - y_2\|^2)$ (with variance at most 2) and that $\|A\|$ is between $\frac{1}{2}\sqrt{d}$ and $2\sqrt{d}$ with probability $1 - \exp(-\Omega(d))$ by standard concentration arguments. Thus, W is at most $\eta = \frac{5\sqrt{\log(d)}}{d}$ with probability $1 - \exp(-\Omega(d))$.

Given $A = a$, we can write the output $b = b_1 + b_2$ as a sum of a component b_1 parallel to a and a component b_2 orthogonal to a. Recalling the notation $B = B_1 + B_2$ from the definition of (A, B), we get the following distributions when $\mathsf{side} = 0$:

$$\begin{aligned} b_2 =& \left(\langle \tfrac{a}{\|a\|}, x_2\rangle + \eta\sigma Z\right)\tfrac{a}{\|a\|} \text{ where } Z \sim N(0,1) \text{ and} \\ b_1 =& \Pi(x_2) + B_2 \text{ where } \Pi \text{ is the projector onto the subspace orthogonal to } a. \end{aligned}$$

We get the same distribution with $\mathsf{side} = 1$, except that x_2 is replaced by y_2. Conditioned on a, we have additive noise with a well-understood distribution in both cases. The likelihood ratio thus depends only on W and $\Pi(x_2 - y_2)$.

The first component consists of adding noise with standard deviation $\eta\sigma$ to an input with sensitivity $|W| \leq \frac{5\sqrt{\log(d)}}{d}$; the second consists of adding noise in with standard deviation σ (in all $d - 1$ relevant dimensions) to an input with sensitivity at most 2. Each of these satisfy (ε, δ)-differential privacy for $\varepsilon = \Theta(\sqrt{\ln(1/\delta)}/\sigma)$, as desired. ☐

Proposition D.2. *When $\sigma\eta < 1/3$, the mechanism $\mathcal{M}$ is not adaptively $(\varepsilon, \frac{1}{4})$-DP unless $\varepsilon \geq \frac{1}{3(\sigma\eta)^2}$.*

Proof. An adaptive adversary first submits vectors x_1, y_1 (both 0), receives a first output a which is either $x_1 + A$ or $y_1 + A$, and then submits x_2 and y_2 and receives a second output b which is either $x_2 + B$ or $y_2 + B$. Consider the specific adversary submits $x_2 = \frac{a}{\|a\|}$ and $y_2 = -x_2$ (based on the first output a) and then receives output b.

The idea is that the variance of B in the direction of $x_2 = a$ is only $\eta\sigma$ (instead of σ) and so—informally—the effective ε of the mechanism is roughly $1/(\eta\sigma)$ instead of $1/\sigma$. When d is large, η is much smaller than 1 and so the mechanism provides much weaker privacy guarantees in the adaptive setting.

More formally, consider the random variable $\langle a, b\rangle$. The component of B in the direction of a can be written $B_2 = \eta\frac{a}{\|a\|} \cdot Z$ for $Z \sim N(0,1)$. When $\mathsf{side} = 0$, we thus have

$$\langle a, b\rangle = \langle a, x_2 + \sigma B\rangle = \langle a, x_2 + \sigma B_2\rangle = \langle a, \tfrac{a}{\|a\|} + \sigma\eta\tfrac{a}{\|a\|}Z\rangle = \|a\|\left(1 + \sigma\eta Z\right).$$

Similarly, when $\mathsf{side} = 1$, the inner product $\langle a, b\rangle$ is distributed as $\|a\|\,(-1 + \sigma\eta Z)$. The probability that $\langle a, b\rangle > 0$ is at least $\frac{1}{2}$ when $\mathsf{side} = 1$ and, for $\sigma\eta < 1$, the same probability is at most $\exp\left(-\frac{1}{2(\sigma\eta)^2}\right)$ when $\mathsf{side} = 0$. In particular, the mechanism is not (ε, δ)-DP in the adaptive model unless $\frac{1}{2} \leq e^{\varepsilon}\Pr(\langle a, b\rangle > 0 | \mathsf{side} = 0) - \delta$; that is, it requires $\varepsilon \geq \frac{1}{2(\sigma\eta)^2} - \ln(\frac{2}{1-2\delta})$. The bound is at least $\frac{1}{3(\sigma\eta)^2}$ for $\delta \leq 1/4$ and $\sigma\eta < 1/3$. ☐

E Proofs and observations for Section 3

E.1 Proof of Theorem 3.1

Proof. The proof essentially follows from standard arguments about the DP guarantee for the Gaussian mechanism [32, 35]. In the following, we provide some of the details for completeness.

First, notice that it is sufficient to state that the computation $\mathbf{CG} + \mathbf{Z}$ satisfies (ε, δ)-DP, due to the post processing property of DP. Now consider two data sets $\mathbf{G}$ and $\mathbf{H}$ differing in one data record (as per the neighborhood definition in J.1). We have $\mathbf{C}(\mathbf{G} - \mathbf{H})$ is equal to the outer product $\mathbf{cg}$, where $\mathbf{g}$ is the row of $\mathbf{G}$ that was changed, and $\mathbf{c}$ is the corresponding column of $\mathbf{C}$. By assumption in the theorem statement, we have

$$\|\mathbf{cg}\|_F \leq \|\mathbf{c}\|_2 \cdot \|\mathbf{g}\|_2 \leq \gamma\zeta.$$

With the bound on the sensitivity above, if each entry of $\mathbf{Z}$ is drawn i.i.d. from $\mathcal{N}(0, \sigma^2)$, then $\mathbf{CG} + \mathbf{Z}$ satisfies $\rho = \frac{\gamma^2\zeta^2}{\sigma^2}$-zCDP (Definiton J.2) [33]. We set the noise standard deviation σ by the use of Remark 15 in Steinke [58]. Correspondingly, we have (ε, δ)-DP [59]. ☐

E.2 Proof of Theorem 3.2

Proof. For simplicity we consider the equality-constrained version of Eq. (4) (permissible by [27]):

$$\mathbf{X}^\star = \underset{\mathbf{X} \text{ is PD}, \mathbf{X}_{[i,i]}=1, i\in[n]}{\arg\min} \operatorname{tr}(\mathbf{A}^*\mathbf{A}\mathbf{X}^{-1}). \tag{13}$$

We begin by noting that Slater's condition (see [60, Section 5.2.3]) holds in our setting, since the minimum eigenvalue of a matrix is a concave function (expressible as a minimum of linear functions), and we know from [27] that that the optimum is strictly positive definite. Therefore strong duality holds, and complementary slackness implies we may drop the positive-definiteness constraint when we move to the Lagrange formulation. Thus, we introduce a Lagrange multiplier $\mathbf{v}$ for Eq. (13), defining,

$$\begin{aligned} L(\mathbf{X}, \mathbf{v}) &= \operatorname{tr}(\mathbf{A}^*\mathbf{A}\mathbf{X}^{-1}) + \sum_{i=1}^{n} \mathbf{v}_i(\mathbf{X}_{i,i} - 1) \\ &= \operatorname{tr}(\mathbf{A}^*\mathbf{A}\mathbf{X}^{-1}) + \operatorname{tr}\big(\operatorname{diag}(\mathbf{v})(\mathbf{X} - \mathbf{I})\big). \end{aligned} \tag{14}$$

Differentiating Eq. (14) with respect to $\mathbf{X}$, we find

$$\frac{\partial L}{\partial \mathbf{X}} = -(\mathbf{X}^{-1}\mathbf{A}^*\mathbf{A}\mathbf{X}^{-1}) + \operatorname{diag}(\mathbf{v}). \tag{15}$$

Let $\mathbf{X}^\star$ be the optimizer of the primal problem Eq. (13); then, by the Lagrange multiplier theorem (see, e.g., Proposition 4.1.1 of [61]), there exists a unique $\mathbf{v}^\star$ satisfying

$$\operatorname{diag}(\mathbf{v}^\star) = \mathbf{X}^{\star-1}\mathbf{A}^*\mathbf{A}\mathbf{X}^{\star-1}, \tag{16}$$

which is an equivalent form of Eq. (7). Since $\mathbf{A}$ is full-rank, and $\mathbf{X}^\star$ is known from [27] to be positive-definite, Eq. (16) implies that $\operatorname{diag}(\mathbf{v})$ is invertible (indeed, positive definite).

Solving Eq. (16) for $\mathbf{X}$ corresponds to solving for a generalized matrix square root. The equation Eq. (16) may be uniquely solved, yielding

$$\mathcal{X}(\mathbf{v}) = \operatorname{diag}(\mathbf{v})^{-1/2}\left(\operatorname{diag}(\mathbf{v})^{1/2}\mathbf{A}^*\mathbf{A}\operatorname{diag}(\mathbf{v})^{1/2}\right)^{1/2}\operatorname{diag}(\mathbf{v})^{-1/2}. \tag{6}$$

Clearly the $\mathcal{X}(\mathbf{v}^\star)$ defined by Eq. (6) represents a solution for Eq. (16); that $\mathcal{X}(\mathbf{v}^\star) = \mathbf{X}^\star$ can be seen by substituting Eq. (16) for $\operatorname{diag}(\mathbf{v}^\star)$ in Eq. (6), and evaluating the result to the form $\mathbf{X}^\star$.

Since $\mathbf{X}^\star$ has constant 1s on the diagonal (by the formulation Eq. (13)), the expression Eq. (6) implies that

$$\operatorname{diagpart}\left(\sqrt{\operatorname{diag}(\mathbf{v}^\star)^{1/2}\mathbf{A}^*\mathbf{A}\operatorname{diag}(\mathbf{v}^\star)^{1/2}}\right) = \mathbf{v}^\star,$$

and that therefore $\mathbf{v}^\star$ is a fixed point of the mapping ϕ defined by Eq. (5).

We have shown that an optimizer $\mathbf{X}^\star$ corresponds to a fixed point $\mathbf{v}^\star$ of ϕ. If we begin with a fixed point $\mathbf{v}^\star$ of ϕ, and define $\mathcal{X}(\mathbf{v}^\star)$ via Eq. (6), the preceding calculations show that $\mathcal{X}(\mathbf{v}^\star)$ is both feasible and a stationary point of the Lagrangian. The strict convexity of the problem, along with its smoothness, imply that the Hessian of the Lagrangian is positive definite at this stationary point, and therefore (e.g. by Proposition 4.2.1 of [61]), this $\mathcal{X}(\mathbf{v}^\star)$ is a local minimizer. Strict convexity implies that this local minimizer is in fact the global minimizer.

The final claim of Eq. (8) follows immediately from weak duality and the fact that

$$\inf_{\mathbf{X}} L(\mathbf{X}, \mathbf{v}) = L(\mathcal{X}(\mathbf{v}), \mathbf{v}))$$

since the problem on the left is convex in $\mathbf{X}$, and hence Eq. (6) gives an optimality condition. Eq. (8) follows by using $\mathbf{A}^*\mathbf{A}\mathbf{X}^{-1} = \mathbf{X}\,\mathrm{diag}\,(\mathbf{v}^\star)$ in the first trace in Eq. (14), and then simplifying using properties of the matrix trace. ☐

E.3 Proof of local-contractive property of ϕ.

Recall that we study the map, defined in Eq. (5),

$$\phi(\mathbf{v}) := \mathrm{diagpart}\left(\sqrt{\mathrm{diag}(\mathbf{v})^{1/2}\,\mathbf{B}^*\mathbf{B}\;\mathrm{diag}(\mathbf{v})^{1/2}}\right),$$

from the positive cone in $\mathbb{R}^n$ to itself. By Theorem 3.2, we know that it has a unique fixed point, which we denote by $\mathbf{v}^\star$. We will need some notation:

- $\mathbf{Q} = \sqrt{\mathbf{B}^*\mathbf{B}}$.
- In $\mathbb{R}^n$, we consider two inner products

  $$\langle \mathbf{x}, \mathbf{y}\rangle = \sum_j x_j y_j, \quad \langle \mathbf{x}, \mathbf{y}\rangle_1 = \sum_j x_j y_j w_j, \quad w_j^{-1} = v_j^*\,.$$

  The fist one is Euclidean, the second one is weighted with the weight given by $\mathbf{v}^\star$ itself. The corresponding norms are $\|\mathbf{x}\|$ and $\|\mathbf{x}\|_1$.
- The operator norms of linear map $\mathbf{A}$ acting in $\mathbb{R}^n$ will be denoted

  $$\|\mathbf{A}\|,\ \|\mathbf{A}\|_1$$

  depending on the considered inner products, e.g.,

  $$\|\mathbf{A}\|_1 = \sup_{\mathbf{x}:\|\mathbf{x}\|_1=1} \|\mathbf{A}\mathbf{x}\|_1\,.$$

We start by giving a simple estimate on $\mathbf{v}^\star$.

Proposition E.1. *Suppose* $\mathbf{Q}$ *satisfies*

$$0 < \kappa_1 \le \mathbf{Q} \le \kappa_2 \tag{17}$$

with some constants κ_1 *and* κ_2*. Then,*

$$\kappa_1^2 \le \mathrm{diag}\,\mathbf{v}^\star \le \kappa_2^2\,.$$

Proof. Given any two non-negative matrices $\mathbf{A}$ and $\mathbf{B}$ that satisfy $\mathbf{A} \le \mathbf{B}$, we clearly have

$$\mathrm{diagpart}\,\mathbf{A} \le \mathrm{diagpart}\,\mathbf{B}, \quad \sqrt{\mathbf{A}} \le \sqrt{\mathbf{B}}\,. \tag{18}$$

Then,

$$\kappa_1^2 \le \mathbf{Q}^2 \le \kappa_2^2 \Rightarrow \kappa_1^2(\mathrm{diag}\,\mathbf{v}^\star) \le (\mathrm{diag}\,\mathbf{v}^\star)^{1/2}\mathbf{Q}^2(\mathrm{diag}\,\mathbf{v}^\star)^{1/2} \le \kappa_2^2(\mathrm{diag}\,\mathbf{v}^\star)\,.$$

Thus, we apply (18) by first taking the square roots and then the diagonal parts of both sides to get

$$\kappa_1\sqrt{\mathrm{diag}\,\mathbf{v}^\star} \le \mathrm{diag}\,\mathbf{v}^\star \le \kappa_2\sqrt{\mathrm{diag}\,\mathbf{v}^\star}$$

after we recall that $\mathbf{v}^\star$ is the fixed point of $\phi(\mathbf{v})$. The required statement is now immediate. ☐

Remark. The argument in the proof shows that ϕ maps the convex set $\{\mathbf{v} : \alpha \leq \operatorname{diag} \mathbf{v} \leq \beta\}$ into itself provided that $0 < \alpha \leq C_1$ and $C_2 \leq \beta$. Since ϕ is continuous, the Brouwer fixed point theorem gives yet another proof that a fixed point of ϕ exists.

The map $\mathbf{v} \mapsto \phi(\mathbf{v})$ is smooth on $\mathbb{R}^n$. Its derivative at point $\mathbf{v}^\star$ is therefore a linear map in $\mathbb{R}^n$. We will denote

$$\mathbf{L} := D\phi(\mathbf{v}^\star). \tag{19}$$

Our central result is precisely the statement that the (weighted) norm of $\mathbf{L}$ is smaller than 1, and hence ϕ is a local contraction around $\mathbf{v}^\star$:

Theorem E.1. *The map L is a contraction in weighted norm, i.e.,*

$$\|\mathbf{L}\|_1 \leq C(\kappa_1, \kappa_2) < 1\,.$$

Remark. This immediately implies that the sequence $\{\mathbf{v}_n\}$ given by $\mathbf{v}_{n+1} = \phi(\mathbf{v}_n)$ converges exponentially fast to $\mathbf{v}^\star$ when $\mathbf{v}_0$ is chosen sufficiently close to $\mathbf{v}^\star$. The exact parameters here depend only on κ_1 and κ_2. The size of the neighborhood in which the first-order approximation implies that ϕ itself is a contraction similarly depends on κ_1, as this controls the smoothness of ϕ.

We will recall some facts before giving the proof of this theorem.

Proposition E.2. *If $\mathbf{A}$ is $n \times n$ matrix and $\mathbf{d} \in \mathbb{R}^\mathbf{n}$, then*

$$\left(\mathbf{A} + \frac{1}{2}(\operatorname{diag} \mathbf{d})\mathbf{A} + \frac{1}{2}\mathbf{A}(\operatorname{diag} \mathbf{d})\right)^2 = \mathbf{A}^2(\operatorname{diag} \mathbf{d}) + (\operatorname{diag} \mathbf{d})\mathbf{A^2} + \mathbf{A}(\operatorname{diag} \mathbf{d})\mathbf{A} + \mathbf{O}(\|\mathbf{d}\|^\mathbf{2}) \tag{20}$$

where the constants in O depend on $\|\mathbf{A}\|$ only.

Proof. That is an immediate calculation. □

Proposition E.3. *If $\mathbf{A}$ is $n \times n$ positive matrix, then*

$$\sqrt{\mathbf{A}} = \frac{\mathbf{A}}{\pi} \int_0^\infty (\mathbf{A} + t)^{-1} \frac{dt}{\sqrt{t}}\,. \tag{21}$$

Proof. That follows from the Spectral Theorem for symmetric matrices and the trigonometric integral formula

$$\sqrt{\lambda} = \frac{\lambda}{\pi} \int_0^\infty (\lambda + t)^{-1} \frac{dt}{\sqrt{t}}, \quad \lambda > 0\,,$$

which follows by substituting $t = \tan^2 \theta$. □

Proposition E.4. *If $\mathbf{A}, \mathbf{V}$ are $n \times n$ matrices and both $\mathbf{A}$ and $\mathbf{A} + \mathbf{V}$ are non-degenerate, then*

$$(\mathbf{A} + \mathbf{V})^{-1} = \mathbf{A}^{-1} - (\mathbf{A} + \mathbf{V})^{-1}\mathbf{V}\mathbf{A}^{-1}$$

and

$$(\mathbf{A} + \mathbf{V})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{V}\mathbf{A}^{-1} + O(\|\mathbf{V}\|^2). \tag{22}$$

Proof. To check the first identity, it is enough to multiply it from the left by $\mathbf{A} + \mathbf{V}$ and from the right by $\mathbf{A}$. The second identity will follows by iterating the first identity once. □

The formula for $\mathbf{L}$ is given in the following lemma.

Lemma E.1. *If $\mathbf{T} := \sqrt{(\operatorname{diag} \mathbf{v}^\star)^{1/2}\mathbf{Q}^2(\operatorname{diag} \mathbf{v}^\star)^{1/2}}$, then*

$$\mathbf{L}\mathbf{w} = \mathbf{w} - \pi^{-1} \operatorname{diagpart}\left(\int_0^\infty (\mathbf{T}^2 + t)^{-1}\mathbf{T}(\operatorname{diag} \mathbf{w})(\operatorname{diag} \mathbf{v}^\star)^{-1}\mathbf{T}(\mathbf{T}^2 + t)^{-1}\sqrt{t}dt\right)\,.$$

Proof. This result is based on a long but straightforward calculation. First, introduce $\mathbf{d} \in \mathbb{R}^n$ by

$$\operatorname{diag}\mathbf{v} = (\operatorname{diag}\mathbf{v}^\star)\exp(\operatorname{diag}\mathbf{d})\,.$$

Hence, denoting $\boldsymbol{\Delta} := \operatorname{diag}\mathbf{d}$ for shorthand, one has

$$\phi(\mathbf{v}) = \operatorname{diagpart}\left(\sqrt{\mathbf{T}^2 + 0.5\boldsymbol{\Delta}\mathbf{T}^2 + 0.5\mathbf{T}^2\boldsymbol{\Delta}}\right) + O(\|\boldsymbol{\Delta}\|^2)$$

since $\mathbf{T} > 0$ and the matrix square-root is Lipschitz-continuous at every point which represents a positive matrix. If one denotes $\mathbf{X} := \sqrt{\mathbf{T}^2 + 0.5\boldsymbol{\Delta}\mathbf{T}^2 + 0.5\mathbf{T}^2\boldsymbol{\Delta}}$, then Propositions E.2 and E.3 yield

$$\begin{gathered}\mathbf{X} - (\mathbf{T} + 0.5\boldsymbol{\Delta}\mathbf{T} + 0.5\boldsymbol{\Delta}\mathbf{T}) = \\ \frac{\mathbf{X}^2}{\pi}\int_0^\infty (\mathbf{X}^2 + t)^{-1}\frac{dt}{\sqrt{t}} - \frac{\mathbf{X}^2 + \mathbf{T}\boldsymbol{\Delta}\mathbf{T}}{\pi}\int_0^\infty (\mathbf{X}^2 + \mathbf{T}\boldsymbol{\Delta}\mathbf{T} + t)^{-1}\frac{dt}{\sqrt{t}} = \\ \overset{(22)}{=} \frac{\mathbf{T}^2}{\pi}\int_0^\infty (\mathbf{T}^2 + t)^{-1}\mathbf{T}\boldsymbol{\Delta}\mathbf{T}(\mathbf{T}^2 + t)^{-1}\frac{dt}{\sqrt{t}} - \mathbf{T}\boldsymbol{\Delta} + O(\|\boldsymbol{\Delta}\|^2).\end{gathered}$$

If we recall that $\mathbf{v}^\star = \phi(\mathbf{v}^\star) = \operatorname{diagpart}\mathbf{T}$, then

$$\begin{gathered}\operatorname{diagpart}\mathbf{X} = \operatorname{diagpart}(\mathbf{T} + 0.5\boldsymbol{\Delta}\mathbf{T} - 0.5\mathbf{T}\boldsymbol{\Delta}) + \\ \operatorname{diagpart}\frac{\mathbf{T}^2}{\pi}\int_0^\infty (\mathbf{T}^2 + t)^{-1}\mathbf{T}\boldsymbol{\Delta}\mathbf{T}(\mathbf{T}^2 + t)^{-1}\frac{dt}{\sqrt{t}} + O(\|\boldsymbol{\Delta}\|^2) = \\ \operatorname{diagpart}\mathbf{T} + \operatorname{diagpart}\frac{1}{\pi}\int_0^\infty (\mathbf{T}^2 + t - t)(\mathbf{T}^2 + t)^{-1}\mathbf{T}\boldsymbol{\Delta}\mathbf{T}(\mathbf{T}^2 + t)^{-1}\frac{dt}{\sqrt{t}} + O(\|\boldsymbol{\Delta}\|^2) = \\ \phi(\mathbf{v}^\star) + \operatorname{diagpart}(\mathbf{T}\boldsymbol{\Delta}) - \pi^{-1}\operatorname{diagpart}\left(\int_0^\infty (\mathbf{T}^2 + t)^{-1}\mathbf{T}\boldsymbol{\Delta}\mathbf{T}(\mathbf{T}^2 + t)^{-1}\sqrt{t}dt\right) + O(\|\boldsymbol{\Delta}\|^2) = \\ \phi(\mathbf{v}^\star) + \operatorname{diagpart}((\operatorname{diag}\mathbf{v}^\star)\boldsymbol{\Delta}) - \pi^{-1}\operatorname{diagpart}\left(\int_0^\infty (\mathbf{T}^2 + t)^{-1}\mathbf{T}\boldsymbol{\Delta}\mathbf{T}(\mathbf{T}^2 + t)^{-1}\sqrt{t}dt\right) + O(\|\boldsymbol{\Delta}\|^2)\end{gathered} \tag{23}$$

Now, notice that

$$\operatorname{diag}\mathbf{v} = \operatorname{diag}\mathbf{v}^\star + \operatorname{diag}\mathbf{v}^\star\boldsymbol{\Delta} + O(\|\boldsymbol{\Delta}\|^{\mathbf{2}})$$

and

$$\boldsymbol{\Delta} = (\operatorname{diag}\mathbf{w})(\operatorname{diag}\mathbf{v}^\star)^{-1} + O(\|\boldsymbol{\Delta}\|^{\mathbf{2}})$$

where $\mathbf{w} := \mathbf{v} - \mathbf{v}^\star$. Finally, we have

$$\begin{aligned}&\phi(\mathbf{v}) = \phi(\mathbf{v}^\star) + \mathbf{w} - \\ &\quad\frac{1}{\pi}\operatorname{diagpart}\left(\int_0^\infty (\mathbf{T}^2 + t)^{-1}\mathbf{T}(\operatorname{diag}\mathbf{w})(\operatorname{diag}\mathbf{v}^\star)^{-1}\mathbf{T}(\mathbf{T}^2 + t)^{-1}\sqrt{t}dt\right) + O(\|\mathbf{w}\|^2)\end{aligned} \tag{24}$$

and that proves the required statement. □

Our next step is to obtain the matrix representation of $\mathbf{L}$ in the standard basis of $\mathbb{R}^n$.

Lemma E.2. *If* $(\mathbf{T}^2 + t)^{-1}\mathbf{T} =: \mathbf{C}(t) = \mathbf{C}_{[i,j]}(t)$, *then*

$$\mathbf{L} = \mathbf{I} - \left\{\frac{1}{\pi}\int_0^\infty (\mathbf{C}_{[i,j]}(t))^2\frac{\sqrt{t}}{v_j^*}dt\right\}, \quad i, j \in \{1, \ldots, n\}\,.$$

Proof. That calculation is straightforward after we use symmetry of matrices $\mathbf{T}$ and $\mathbf{C}$. □

The Theorem E.1 will be proved if Lemma E.3 is shown. In its proof, the following property of the Schur (elementwise, also known as Hadamard) product of two matrices is used.

Proposition E.5. *Suppose* $\mathbf{A}$ *and* $\mathbf{B}$ *are non-negative matrices of size* $n \times n$. *Then,*

$$\lambda_{\min}(\mathbf{A} \circ \mathbf{B}) \geq \lambda_{\min}(\mathbf{A})\lambda_{\min}(\mathbf{B})\,.$$

Proof. Indeed, the matrix $\mathbf{A} \circ \mathbf{B}$ represents the principal submatrix of the Kronecker (or tensor) product $\mathbf{A} \otimes \mathbf{B}$. Since $\lambda_{\min}(\mathbf{A} \otimes \mathbf{B}) = \lambda_{\min}(\mathbf{A})\lambda_{\min}(\mathbf{B})$, we get our result. $\square$

Lemma E.3. *The operator* $\mathbf{L}$ *is selfadjoint with respect to the weighted inner product and* $\|\mathbf{L}\|_1 \le C(\kappa_1, \kappa_2) < 1$.

Proof. First, we can write a bilinear form

$$\langle \mathbf{L}\mathbf{v}, \mathbf{w} \rangle_1 = \langle \mathbf{v}, \mathbf{w} \rangle_1 - \frac{1}{\pi} \int_0^\infty \sum_{i,j=1}^n \mathbf{G}_{[i,j]}(t) \frac{v_j w_i}{v_j^* v_i^*} \sqrt{t} dt$$

where $\mathbf{G} := \mathbf{C} \circ \mathbf{C}$, the Schur product, is a symmetric matrix. Hence, $\langle \mathbf{L}\mathbf{v}, \mathbf{w} \rangle_1 = \langle \mathbf{L}\mathbf{w}, \mathbf{v} \rangle_1$ and therefore $\mathbf{L}$ is appropriately selfadjoint. Next, we will prove a bound

$$C(\kappa_1, \kappa_2) \|\mathbf{v}\|_1^2 \le \frac{1}{\pi} \int_0^\infty \sum_{i,j=1}^n \mathbf{G}_{[i,j]}(t) \frac{v_j v_i}{v_j^* v_i^*} \sqrt{t} dt \le C_5 \|\mathbf{v}\|_1^2 \tag{25}$$

with some positive C and $C_5 \in (0, 1)$. That estimate for quadratic form is sufficient to prove that $\|\mathbf{L}\|_1 < 1$ due to the variational characterization of the norm of a self-adjoint operator, i.e., $\|\mathbf{L}\|_1 = \sup_{\mathbf{v}: \|\mathbf{v}\|_1 = 1} |\langle \mathbf{L}\mathbf{v}, \mathbf{v} \rangle_1|$.

We claim that

$$\int_0^\infty \mathbf{G}(t) \sqrt{t} dt \ge C_3(\kappa_1, \kappa_2) > 0 \tag{26}$$

in a sense of positive matrices. Indeed,

$$\lambda_{\min}(\mathbf{G}(t)) \ge (\lambda_{\min}(\mathbf{C}(t)))^2$$

as follows from the properties of the Schur product. Since $\mathbf{C}(t) = \mathbf{T}/(\mathbf{T}^2 + t)$, we get

$$\int_0^\infty (\lambda_{\min}(\mathbf{C}(t)))^2 \sqrt{t} dt \ge C_3(\kappa_1, \kappa_2) > 0$$

where C_3 depends on parameters κ_1 and κ_2 from (17) only. So, our claim (26) is proved. Given (26), we can write

$$\int_0^\infty \sum_{i,j=1}^n \mathbf{G}_{[i,j]}(t) \frac{v_j v_i}{v_j^* v_i^*} \sqrt{t} dt \ge C_3(\kappa_1, \kappa_2) \sum_{j=1}^n \left| \frac{v_j}{v_j^*} \right|^2 \ge$$

$$C_4(\kappa_1, \kappa_2) \sum_{j=1}^n \frac{|v_j|^2}{|v_j^*|} = C_4(\kappa_1, \kappa_2) \|\mathbf{v}\|_1^2$$

thanks to Proposition E.1. This shows the left bound in Eq. (25).

The inequality

$$\frac{1}{\pi} \sum_{i,j=1}^n \mathbf{G}_{[i,j]}(t) \frac{v_j v_i}{v_j^* v_i^*} \sqrt{t} dt \le C_5 \|\mathbf{v}\|_1^2$$

is equivalent to

$$\frac{1}{\pi} \sum_{i,j=1}^n \mathbf{G}_{[i,j]}(t) \frac{x_j x_i}{\sqrt{v_j^* v_i^*}} \sqrt{t} dt \le C_5 \|\mathbf{x}\|^2 \tag{27}$$

if we make the change of variables $x_j := v_j / \sqrt{v_j^*}, j \in \{1, \ldots, n\}$. It will be convenient to introduce a symmetric matrix $\mathbf{D}$ with coefficients given by

$$\mathbf{D}_{[i,j]} = \frac{1}{\pi} \int_0^\infty \mathbf{G}_{[i,j]}(t) \frac{1}{\sqrt{v_j^* v_i^*}} \sqrt{t} dt = \frac{1}{\pi} \int_0^\infty (\mathbf{C}_{[i,j]}(t))^2 \frac{1}{\sqrt{v_j^* v_i^*}} \sqrt{t} dt \,.$$

To bound the norm of this matrix, we will start with the following observation. The application of Spectral Theorem to matrix $\mathbf{T}$ yields

$$\frac{1}{\pi} \int_0^\infty (\mathbf{T}^2 + t)^{-2} \mathbf{T}^2 \sqrt{t} dt = C_5 \mathbf{T}$$

where

$$C_5 = \frac{1}{\pi}\int_0^\infty (1+\xi)^{-2}\sqrt{\xi}d\xi = \frac{2}{\pi}\int_0^\infty (1+u^2)^{-2}u^2 du < \frac{2}{\pi}\int_0^\infty (1+u^2)^{-1}du = 1\,.$$

Recall also that $\mathbf{v}^\star = \operatorname{diagpart}\mathbf{T}$ and therefore

$$\frac{1}{\pi}\int_0^\infty \operatorname{diagpart}((\mathbf{T}^2+t)^{-2}\mathbf{T}^2)\sqrt{t}dt = C_5\mathbf{v}^\star$$

Since the matrix elements of $\mathbf{C}(t) = (\mathbf{T}^2+t)^{-1}\mathbf{T}$ are given by $\mathbf{C}_{[i,j]}(t)$, the diagonal elements of $(\mathbf{T}^2+t)^{-2}\mathbf{T}^2$ can be obtained by the formula

$$\sum_{j=1}^n \mathbf{C}_{[i,j]}(t)\mathbf{C}_{[j,i]}(t) = \sum_{j=1}^n (\mathbf{C}_{[i,j]}(t))^2$$

for $i \in \{1,\ldots,n\}$. Therefore, we get an identity

$$\frac{1}{\pi}\int_0^\infty \sum_{j=1}^n (\mathbf{C}_{[i,j]}(t))^2\sqrt{t}dt = C_5 v_i^*, \quad i\in\{1,\ldots,n\}$$

which can be rewritten as

$$\sum_{j=1}^n \mathbf{D}_{[i,j]}\sqrt{v_i^* v_j^*} = C_5 v_i^*, \quad i\in\{1,\ldots,n\}\,.$$

The elements $\mathbf{D}_{[i,j]}$ are non-negative and $\mathbf{D}_{[i,j]} = \mathbf{D}_{[j,i]}$. Taking the vector $\{\sqrt{v_i^*}\}$ with positive entries, we rewrite the previous identity as

$$\sum_{j=1}^n \mathbf{D}_{[i,j]}\sqrt{v_j^*} = C_5\sqrt{v_i^*}, \quad i\in\{1,\ldots,n\}\,.$$

The application of Schur's test for the norm of matrix gives $\|\mathbf{D}\| \le C_5 < 1$. Since $\mathbf{D}$ is symmetric, this bound implies (27).

□

E.4 Numerical observations of the map ϕ.

Some care must be taken with floating point issues in the implementation of the map ϕ. In particular, numerical evaluation of ϕ depends critically on the computation of a matrix square root, and precision in this computation is crucial for the usability of these fixed-point methods.

Several numerical approaches can improve the stability of these algorithms. In particular, some of the expressions above (e.g., the definition of $\mathcal{X}$) imply a priori lower bounds on the eigenvalues of matrices for which we need square roots. The results of [62] yield straightforward lower bounds that can stabilize our iterative algorithms. These bounds can be applied to ensure the iterates never encounter pathological numerical artifacts. As the size of matrices scales up (in particular, our factorizations usually focused on 2048×2048 matrices, and larger matrices are of interest), we observed that performing all computations in `float64` precision was crucial to minimizing these numerical artifacts.

We observed experimentally that while factorizing some matrices, though the fixed-point method itself converged independently of the matrix factorized, some oscillation in the values of the loss Eq. (2) occurred. Further investigation is needed to determine whether this oscillation represents a true feature of the iterated dynamics, or simply another numerical artifact, due e.g. to lack of precision in the matrix square root. If the former, certain approaches to prove global convergence of these iterates are ruled out: in particular, those which rely on this loss as a potential function, which iterating ϕ always decreases.

To evaluate empirical usefulness of this fixed-point method, we implemented three different algorithms for computing optimal factorizations:

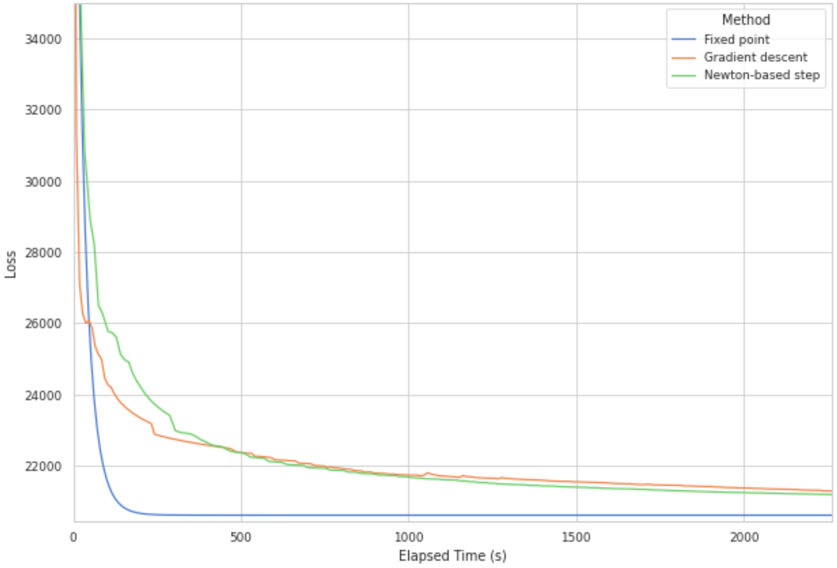

Figure 4: Value of loss Eq. (2) against elapsed time for a gradient-descent based, Newton-direction based, and fixed-point implementation of computing optimal factorizations of 2048-dimensional prefix sum matrix $\mathbf{S}$. The gradient descent and Newton direction-based methods used an Armijo step size search, and checked for existence of Cholesky factorization to verify positive-definiteness of the iterates, as suggested by [27]. The methods were initialized identically, leveraging the expression Eq. (6), a significantly better initialization for the gradient-based methods than might be obvious in the absence of this expression (e.g., initialization to $\mathbf{I}$).

1. A gradient-descent-based procedure to compute the optima of Eq. (4), guaranteed to be convergent by the convexity of the problem.
2. [27, Algorithm 1], a Newton-direction-based algorithm with global convergence guarantees, hand-optimized with the structure of the problem—in particular, avoiding the need to materialize a Hessian with n^4 elements. This implementation used the default settings from [27].
3. Simply iterating the mapping ϕ.

In all situations we tested, the fixed-point method was significantly faster than either of the other two, up to two orders of magnitude in some cases. In Fig. 4, we plot loss against time for an example of 2048×2048 matrix factorization using CPUs. The methods are all similarly amenable to GPU acceleration.

F Proofs for Section 4

F.1 Proof of Proposition 4.1

Proof. Following the analysis of Theorem C.1, Kairouz et al. [6], we introduce a hypothetical 'unnoised' model trajectory $\widetilde{\theta}_t$. Define $b_t := \theta_t - \widetilde{\theta}_t$, and note that $b_t = -\eta \mathbf{B}_{[t,:]}\mathbf{Z}$.

We note the well-known equivalence of FTRL and gradient descent, with requilarization parameter λ equivalent to $\frac{1}{\eta}$ (as can be seen by solving for the FTRL update). Following the standard linearization method of online convex optimization, we see:

$$\frac{1}{n}\sum_{t=1}^{n}\ell(\theta_t;\chi_t) - \ell(\theta^\star;\chi_t) \le \frac{1}{n}\sum_{t=1}^{n}\langle\nabla_t, \theta_t - \theta^\star\rangle = \underbrace{\frac{1}{n}\sum_{t=1}^{n}\langle\nabla_t, \widetilde{\theta}_t - \theta^\star\rangle}_{\text{A}} + \underbrace{\frac{1}{n}\sum_{t=1}^{n}\langle\nabla_t, \theta_t - \widetilde{\theta}_t\rangle}_{\text{B}}$$

Similarly to Kairouz et al. [6], the term A may be handled with standard online convex optimization techniques, yielding

$$\text{A} \le \eta L^2 + \frac{1}{2\eta n}\left(\|\theta^\star\|_2^2 - \|\theta_1\|_2^2\right),$$

so we are left to evaluate the expectation of B over the noise injected by Algorithm 1. We compute:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{n}\sum_{t=1}^{n}\langle\nabla_t, \theta_t - \widetilde{\theta}_t\rangle\right] &\le \frac{1}{n}\mathbb{E}\left[\sum_{t=1}^{n}\|\nabla_t\|_2\|\theta_t - \widetilde{\theta}_t\|_2\right] && \text{Cauchy-Schwartz} \\
&\le \frac{L}{n}\mathbb{E}\left[\sum_{t=1}^{n}\|b_t\|_2\right] \\
&\le L\mathbb{E}\left[\left(\frac{1}{n}\sum_{t=1}^{n}\|b_t\|_2^2\right)^{1/2}\right] && \text{Jensen's inequality} \\
&\le \frac{L}{\sqrt{n}}\left(\mathbb{E}\left[\sum_{t=1}^{n}\|b_t\|_2^2\right]\right)^{1/2} && \text{Jensen again} \\
&= \frac{L\eta}{\sqrt{n}}\left(\mathbb{E}\left[\|\mathbf{B}\mathbf{Z}\|_F^2\right]\right)^{1/2} && \text{definition of } b_t \\
&= \frac{L\sigma\eta}{\sqrt{n}}\|\mathbf{B}\|_F && \text{evaluating the expectation.}
\end{aligned}$$

Putting together the estimates of A and B yields the result.

□

G Visualization of Optimal Factorizations

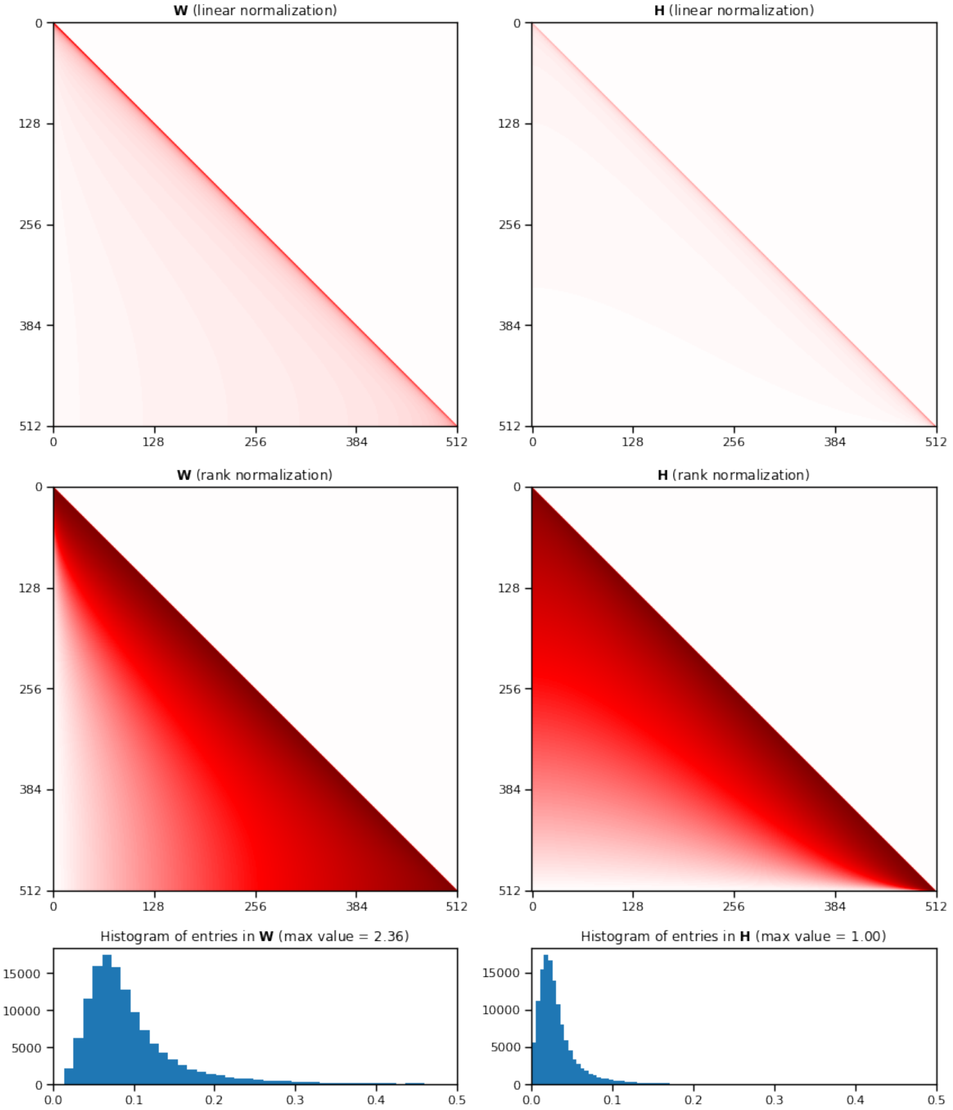

Figure 5: Visualizations of the optimal streaming matrix factorization $\mathbf{S} = \mathbf{WH}$ ($\mathbf{B} = \mathbf{W}, \mathbf{C} = \mathbf{H}$) for cumulative sums with $n = 512$. The matrix visualizations use a color palette that maps scalars in $[0, 1]$ to colors from white to dark red. The first row normalizes entries to $[0, 1]$ by simply dividing all entries by 2.36 (the largest value in either matrix). This clearly shows the heavy diagonal in $\mathbf{B}$. The second row normalizes the values in each matrix by ranking them by magnitude, and then mapping the ranks to $[0, 1]$ so 0 entries (the smallest) are mapped to 0.0, the median value is mapped to 0.5 (mid-red), and the largest value is mapped to 1.0 (darkest red). This visualization more clearly shows the off-diagonal structure. The final row gives a histogram of the magnitudes of the non-zero entries in each matrix.

H Computational efficiency for the matrix mechanism

Our primary goal has been to develop mechanisms with best-possible privacy vs utility tradeoffs in the streaming setting. However, the $(\mathbf{B}, \mathbf{C})$ we compute are in general dense, and do not obviously admit a computationally-efficient implementation of the associated DP mechanism. In contrast, tree aggregation (including, with a careful implementation, the streaming Honaker estimator) allows

n	**Honaker**	$(\mathbf{B}^\star, \mathbf{C}^\star)$	**Efficient**	(h, r)
$2^8 = 256$	74.4	40.4	40.4	(4, 4)
$2^9 = 512$	116.5	62.0	62.2	(5,4)
$2^{10} = 1024$	180.8	94.6	95.5	(5, 5)
$2^{11} = 2048$	278.3	143.6	145.8	(6, 5)
$2^{12} = 4096$	425.6	217.3	224.0	(6, 6)

Table 2: Values of $\sqrt{\mathcal{L}}$ for the expected squared reconstruction error $\mathcal{L}$ defined in Eq. (2) (which implies equivalent levels of privacy). The "Efficient" column gives $\sqrt{\mathcal{L}}$ for the structured approximation $\hat{\mathbf{B}}$ of $\mathbf{B}^\star$ with parameters (h, r) described below. When $n = 2^i$ we choose $h + r = i$, so that the mechanism based on $\hat{\mathbf{B}}$ has memory and computation efficiency comparable to the Honaker approach.

implementations with only $\log(n)$ overhead; that is, each DP estimate of the i^{th} partial sum can be computed in time and space $\mathcal{O}(d \log(n))$.

In this section, we demonstrate empirically that the optimal $\mathbf{B}^\star$ from the factorization of the prefix sum matrix $\mathbf{S}$ can be approximated by structured matrices in such a way as to be competitive with the tree-aggregation approach in terms of computation and memory, but retain the advantage of substantially improved utility. Recalling Algorithm 1, the key is to compute $\mathbf{B}_{[i,:]}\mathbf{Z}$ efficently. If $\mathbf{B}$ is arbitrary, this takes $\mathcal{O}(nd)$ operations, which is likely prohibitive.

However, having a structured matrix $\hat{\mathbf{B}}$ that allows efficient multiplication with $\mathbf{Z}$ mitigates this problem. We propose the following construction, which empirically provides a good approximation while also allowing computational efficiency. Let $\mathbf{D}^{(h)}$ denote the lower-triangular banded matrix formed by taking the first h diagonals of $\mathbf{B}$, so $\mathbf{D}^{(0)}$ is the all-zero matrix, $\mathbf{D}^{(1)}$ is the main diagonal of $\mathbf{B}$, and $\mathbf{D}^{(2)}$ contains the main diagonal and one below it, etc. Let $\mathbf{U}^{(h)} \in \{0,1\}^{n \times n}$ contain a 1 in the place of each non-zero element of $\mathbf{B}$ not captured in $\mathbf{D}^{(h)}$ and zero elsewhere, so in particular $\mathbf{B} = \mathbf{B} \odot \mathbf{U}^{(h)} + \mathbf{D}^{(h)}$ where $\odot$ is elementwise multiplication. Then, we propose the representation

$$\hat{\mathbf{B}} = \left(\mathbf{L}\mathbf{R}^\top\right) \odot \mathbf{U}^{(h)} + \mathbf{D}^{(h)},$$

where $\mathbf{L}, \mathbf{R} \in \mathbb{R}^{n \times r}$. Finding a low-rank factorization $\mathbf{L}\mathbf{R}^\top$ which minimizes $\|\hat{\mathbf{B}} - \mathbf{B}\|_F^2$ can be cast as a matrix completion problem, as we only care about approximating with $\mathbf{L}\mathbf{R}^\top$ the entries of $\mathbf{B}$ selected by $\mathbf{U}^{(h)}$. For these experiments we used an alternating least squares solver with a regularization penalty of 10^{-6} on $\|\mathbf{L}\|_F^2 + \|\mathbf{R}\|_F^2$ [63–65]. Given such a representation, the cost of computing $\hat{\mathbf{B}}_{[i,:]}\mathbf{Z}$ is $\mathcal{O}((h + r)d)$: we maintain accumulators $\boldsymbol{\beta}$ such that

$$(\mathbf{L}\boldsymbol{\beta})_{[i,:]} = \left((\mathbf{L}\mathbf{R}^\top \odot \mathbf{U}^{(h)})\mathbf{Z}\right)_{[i,:]},$$

and $\boldsymbol{\beta}$ can be updated in time rd on each step. Then, Finally, $(\mathbf{D}^{(h)}\mathbf{Z})_{[i,:]}$ can be computed in time hd. Algorithm 3 makes this algorithm explicit.

Columns 3 and 4 in Table 2 shows empirically that this approximation recovers almost all of the accuracy improvement of $(\mathbf{B}^\star, \mathbf{C}^\star)$ at comparable computational efficiency to tree aggregation with the Honaker estimator (that is, we choose $h + r = \log_2(n)$). While a paired $\mathbf{C}$ is not used directly in computing the private estimates, it is necessary in order to compute the loss $\mathcal{L}$ defined in Eq. (2), as well as to appropriately calibrate the noise to achieve a DP guarantee (see Theorem 3.1). For these purposes an optimal $\mathbf{C}_{\hat{\mathbf{B}}}$ can be found analogous to Eq. (3) as $\mathbf{C}_{\hat{\mathbf{B}}} = \hat{\mathbf{B}}^{-1}\mathbf{S}$.

Algorithm 3 An efficient implementation (executed by the trusted curator)

```
# Iterations and matrices/vectors are zero indexed (unlike elsewhere)
Parameters:
        Matrix $\mathbf{D}^{(h)}$ containing $h \in \{0, \dots, n\}$ diagonals from $\mathbf{B}$
        Matrices $\mathbf{L}, \mathbf{R} \in \mathbb{R}^{n \times r}$
        Noise matrix $\mathbf{Z} \in \mathbb{R}^{n \times d}$
        Observations $\mathbf{G} \in \mathbb{R}^{n \times d}$
$\boldsymbol{\beta} := \mathbf{0} \in \mathbb{R}^{r \times d}$        # Buffer for relevant part of $\mathbf{R}^\top \mathbf{Z}$.
$\mathbf{s} := 0 \in \mathbb{R}^d$        # Accumulator for prefix sum
for $i$ in $1, \dots, n$ do
    $\mathbf{s} \mathrel{+}= \mathbf{G}_{[i,:]}$        # Maintain the un-noised cumulative sum
    $\mathbf{y} := 0 \in \mathbb{R}^d$        # Accumulator for total noise in $i$th prefix sum
    for $k$ in $0, \dots, \min(i, h-1)$ do        # Handle $h$ diagonals directly; No-op if $h = 0$
        $\mathbf{y} \mathrel{+}= \mathbf{D}^{(h)}_{[i,i-k]} \mathbf{Z}_{[i-k,:]}$        # $hd$ multiplies
    if $i \geq h$ then        # Compute the low-rank portion
        $i' \leftarrow i - h$
        $\boldsymbol{\beta} \mathrel{+}= \mathbf{R}^\top_{[i',:]} \mathbf{Z}_{[i',:]}$        # $rd$ multiplies
        $\mathbf{y} \mathrel{+}= \mathbf{L}_{[i,:]} \boldsymbol{\beta}$        # $rd$ multiplies
    Release $\mathbf{s} + \mathbf{y}$        # A DP estimate of $\sum_{t=1}^{i} \mathbf{G}_{[t,:]}$
```

I Experiment Details

Mechanism implementation Though Appendix H shows that time- and space-bounded approximations to our optimal factorizations are possible, for our experimental results we followed Algorithm 1 and implemented the straightforward version of our mechanism. That is, we leverage the expression

$$\mathbf{B}\left(\mathbf{C}\mathbf{G} + \mathbf{Z}\right) = \mathbf{A}\mathbf{G} + \mathbf{B}\mathbf{Z},$$

for $\mathbf{A} = \mathbf{B}\mathbf{C}$, where $\mathbf{A}$ represents the linear operator we are interested in estimating. By introducing a seed to the generation of the noise vector $\mathbf{Z}$, the appropriate noise vector $(\mathbf{B}\mathbf{Z})_{[i,:]}$ can simply be computed afresh for each iteration of training (or round in the federated setting). The linear operators $\mathbf{A}$ in which we are interested admit efficient implementations; e.g., gradient descent with momentum can be implemented with a single buffer, representing the current state of the model. The computation of $\mathbf{B}\mathbf{Z}$ is therefore the dominant component in the above.

We normalized all of our factorizations to have sensitivity exactly 1 in the single-pass setting.

Integration with federated learning We implemented these mechanisms via the `DPQuery` interface in TensorFlow-Privacy [66], which integrates naturally with `tff.aggregators`, the aggregators library of TensorFlow-Federated. We were therefore able to reuse precisely the same code for training as [67], simply swapping in our matrix-factorization-based aggregators as an argument to TFF's `tff.learning.build_federated_averaging_process` function. In conjunction with this paper, we are in the process of open-sourcing the code to reproduce our experiments. TFF's distributed C++ runtime, equipped with one machine for every 10 clients per round and low-priority CPU resources, enabled our experiment grids (including evaluation) to finish in approximately 1 day.

Stackoverflow settings The preprocessing of our data, in addition to model architecture as well as the settings of various task-specific hyperparameters like the maximum number of examples processed per-client, we share with [6].

Test accuracy details Test accuracies (excluding predictions on out-of-vocabulary and end-of-sentence tokens) plotted against ε values associated to $\delta = 10^{-6}$ for various instantiations of the mechanisms we tested can be found in Fig. 2. This figure was generated with a sweep over client and server learning rates, with grids chosen via in the heatmap for FedAvgM in Figure 2 of [51], as well as a sweep over server momentum values. The noise multiplier settings were chosen with a simple calculation, based on the reported noise multipliers for StackOverflow NWP in [6]. In particular, by explicitly calculating the sensitivity of the binary tree (as in Theorem 4.1 of [6]), one can normalize the noise multipliers to be equivalent between the two settings. In the process of testing our code, we verified that we observed similar results to those claimed there under this normalization. The smallest noise multiplier in our setting corresponds to the largest ε in figure 2(a) of [6], though our plots are not exactly comparable to theirs due to the different number of rounds in the two experimental setups. We calculate our ε values by simply measuring the privacy cost of the appropriate high-dimensional Gaussian query, by Theorem 4.1. The grid we swept over can be found in Table 3.

The error bars in Figs. 2 and 3 were generated by first filtering down to runs which did not diverge from repeated runs with 10 seeds (at least 7 converged for each setting in Fig. 2, at least 8 for each setting in Fig. 3), then computing the empirical standard deviation. A similar process was used for Fig. 6.

Evaluation accuracy details During training, we monitored performance on an evaluation set consisting of 10,000 sentences from outside of the training and test sets. We plot this evaluation accuracy for the final portion of training our learning-rate schedule and momentum matrices in Fig. 6.

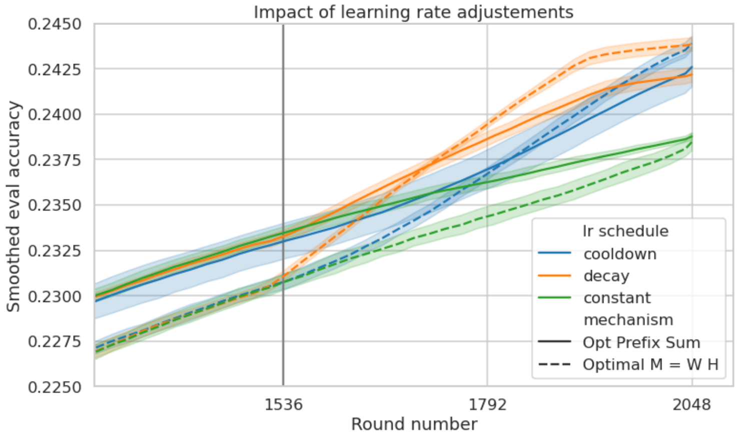

Figure 6: Smoothed validation accuracy over the final 748 rounds at $\varepsilon = 18.9, \delta = 10^{-6}$, comparing momentum and learning rate decay implemented as postprocessing operations to the prefix-sum mechanism versus capturing these operations in the mechanism itself (see Section 4). Vertical line represents the start of the decay schedule.

Additional figures We note that Fig. 6 demonstrates a consistent artifact we witnessed in training these models: the momentum matrix factorization performs worse than the prefix-sum matrix during the body of training, but catches up and overtakes towards the end of the training procedure. We hypothesize this to be an artifact of the way these mechanisms distribute variance on the operator residuals, and consider it an interesting pointer for future mechanism design, while noting that it implies the matrix factorizations are significantly tuned to the number of iterations for which they are designed.

Hyperparameter settings The parameter settings for the various figures in the main body and appendix can be found below.

Table 3: Grids used in initial search for Fig. 2.

Parameter	Grid values
Client learning rate	[0.5, 1.0]
Server learning rate	[0.25, 0.5, 1.0, 2.0]
Server momentum	[0.0, 0.9, 0.95]
Noise multiplier	[0.341, 0.682, 1.364, 2.728, 5.456]

Table 4: Hyperparameter settings for Fig. 2.

Mechanism	ε	**(Server LR, Client LR, Server momentum)**
`Honaker Full`	18.9	$(0.5, 1., 0.95)$
	8.2	$(0.25, 1., 0.95)$
	3.7	$(0.25, 1., 0.9)$
	1.7	$(0.25, 0.5, 0.9)$
	0.8	$(0.25, 0.5, 0.0)$
`Honaker Online`	18.9	$(0.25, 1., 0.95)$
	8.2	$(0.25, 1., 0.9)$
	3.7	$(0.25, 0.5, 0.9)$
	1.7	$(0.25, 1., 0.0)$
	0.8	$(0.25, 0.5, 0.0)$
`Opt Prefix Sum`	18.9	$(0.5, 1., 0.95)$
	8.2	$(0.25, 0.5, 0.95)$
	3.7	$(0.25, 1., 0.9)$
	1.7	$(0.25, 0.5, 0.9)$
	0.8	$(0.5, 0.5, 0.0)$
`Optimal M = B C`	18.9	$(1., 1., 0.9)$
	8.2	$(0.25, 1., 0.95)$
	3.7	$(0.25, 0.5, 0.9)$
	1.7	$(0.25, 0.5, 0.9)$
	0.8	$(0.5, 0.5, 0.0)$

For Fig. 3, the parameter settings differed based on the mechanisms explored. Constant LR schedules used the same settings as $\varepsilon = 18.9$ in Table 4. For the exploration of learning rate decay schedules, a server learning rate of 0.5, client learning rate of 1.0, and server momentum of 0.95 were shared. The plot Fig. 6 was generated from the same set of experiments.

J Background on Differential Privacy

In this paper we operate with the "replace with zero" variant of differential privacy [6, Defn. 2.1], sated below for completeness purposes.

Definition J.1 (Differential privacy)**.** *Let $\mathcal{D}$ be the domain of data records, $\perp \notin \mathcal{D}$ be a special element, and let $\widehat{\mathcal{D}} = \mathcal{D} \cup \{\perp\}$ be the extended domain. A randomized algorithm $\mathcal{A} : \widehat{\mathcal{D}}^n \to \mathcal{S}$ is (ε, δ)-differentially private if for any data set $D \in \widehat{\mathcal{D}}^n$ and any neighbor $D' \in \widehat{\mathcal{D}}^n$ (formed from D by replacing one record with $\perp$), and for any event $S \in \mathcal{S}$, we have*

$$\Pr[\mathcal{A}(D) \in S] \leq e^{\varepsilon} \cdot \Pr[\mathcal{A}(D') \in S] + \delta, \text{ and}$$
$$\Pr[\mathcal{A}(D') \in S] \leq e^{\varepsilon} \cdot \Pr[\mathcal{A}(D) \in S] + \delta,$$

where the probability is over the randomness of $\mathcal{A}$.

In our algorithms, we would treat $\perp$ specially, and assume it corresponds to the all-zeros vector of appropriate dimensions. This definition extends naturally to other variants like Renyi differential privacy (RDP) [35], and zero Concentrated Differential Privacy (zCDP). For completeness purposes we provide the definition of zCDP we primarily use in the paper.

Definition J.2 (zero concentrated differential privacy)**.** *Analogous to the definitiion of (ε, δ)-differential privacy in Definition J.1, a randomized algorithm $\mathcal{A}$ is ρ-zCDP if the condition on*

$\mathcal{A}(D)$ *and* $\mathcal{A}(D')$ *in Definition J.1 are replaced with the following:*

$$\frac{1}{\alpha-1}\log \mathbb{E}_{s\sim\mathcal{A}(D)}\left(\frac{\Pr\left[\mathcal{A}(D)=s\right]}{\Pr\left[\mathcal{A}(D')=s\right]}\right)^{\alpha} \leq \rho\alpha, \quad \textit{and}$$

$$\frac{1}{\alpha-1}\log \mathbb{E}_{s\sim\mathcal{A}(D')}\left(\frac{\Pr\left[\mathcal{A}(D')=s\right]}{\Pr\left[\mathcal{A}(D)=s\right]}\right)^{\alpha} \leq \rho\alpha.$$